Implementation of Serial Port Communication Based on Modbus-RTU Communication Protocol in C#.NET

Pan Zhao*[1], Yong Zheng[2]

College of Mechanical and Electrical Engineering, North China University of Technology, Beijing, China

*[1]277746909@qq.com; [2]cntzy@ncut.edu.cn

Abstract

This paper introduces Modbus-RTU Communication protocol, and combining with an instance, it states the specific method to implement the serial port communication between computer and ADVANTECH 4018+ modules by the Serial Port control based on the analysis of the principle of communication in C#.NET.

Keywords

Modbus-RTU Communication Protocol; C#; Serial Port Control; CRC Check

Introduction

This paper states the instance which uses the Serial Port control and Modbus-RTU Communication protocol of C#, successes to implement the serial port communication function between IPC and distributed measurement modules with RS-485 port in the system of the multichannel temperature and pressure measurement from the perspective of application.

Setting up the Distributed Measurement Module

The project uses eight ADVANCH 4018+ distributed measurement modules. Every module can measure the thermocouple and current signal of eight channels and allows the users to preset the address code. This project runs the Modbus-RTU communication protocol and sets the baud rate 9600bps. FIG.1 shows the hardware of the system.

FIG. 1 THE HARDWARE DESIGN OF THE SYSTEM OF DISTRIBUTED TEMPERATURE AND PRESSURE MEASUREMENT

Introduction of Modbus-RTU Communication Protocol

Modbus-RTU communication protocol uses the half-duplex communication mode. The mainframe sends the command signal to the terminal equipment according to the different slave address. After the corresponding operation, the terminal equipment sends the answering signal to the mainframe. The protocol only permits the communication between mainframe and terminal equipment and does not permit it among terminal equipments. Tabel 1 shows the data frame format of Modbus-RTU communication protocol.

TABLE 1 THE DATA FRAME FORMAT OF MODBUS-RTU COMMUNICATION PROTOCOL

Address Code	Function Code	Data Area	Check Code
8 bits	8 bits	N*8 bits	16 bits

The computer uses the query response mode to

TABLE 2 THE QUERY COMMAND OF THE COMPUTER

Address Code	Function Code	Starting Channel Number (high-order)	Starting Channel Number (low -order)	The Number of Data (high-order)	The Number of Data (low -order)	CRC Low Byte	CRC High Byte
01-08	03	00	00-07	00	01	low byte	high byte

TABLE 3 THE RESPONSE MESSAGE FORMAT OF 4018+ MODULES

Device Number	Function Code	The Number of Bytes in the Data	Data1 High Byte	Data 1 Low Byte	CRC Low Byte	CRC High Byte
01-08	03	02			low byte	High byte

communicate to the ADAM-4018+ modules. The modules send a response message after the computer sends a query command. The query command format of the computer shows in Table 2 according to the Modbus-RTU communication protocol and actual requirement.

Address code "01-08," means the data that computer obtains by querying the 01 to 08 modules; function code "03" means the operation for modules reading the register values; 4018+ is 8-way differential input module. The computer can query the data of 00 to 07 channel. So the high-order of initial channel is "00" and low-order is 00 to 07; data length "00 01" means the channel data for one channel starting from the initial channel; CRC check code is calculated by program.

After receiving the command from computer, the modules execute the corresponding function and return the response message. Table 3 shows the response message format.

Software Design

Usually people use MSCommn communication control to write a serial port communication program in Visual Studio .NET. But Visual Studio .NET does not add this control to the control library in the Microsoft .NET technology today. Since Visual Studio 2005 development tool is launched, in the subsequent versions, .NET class library contains Serial Port class which also can realize the serial communication function and play the same role like MSCommn communication control. Visual Studio 2010 is used in this program which contains Serial Port. C# is an object-oriented programming language which is relative easy to learn. This program implements the serial port communication between computer and ADAM-4018+ modules by Serial Port control in the C# .NET environment.

Implementation of Communication

Serial Port class's namespace is System. IO.Ports, so only adding "using System. IO.Ports"at the beginning of the routine, the class can be used. As with other controls, Serial Port interacts with user by attribute, method and event. Through the design of attribute, method and event, Serial Port can send and receive data.

The general process of serial communication for Serial Port control is opening port, sending data, receiving data and closing port.

1) Initialization and Opening Port

```
Serial Port1.PortName = "COM" + PortCmb. Text;
Serial Port1.BaudRate=
Convert. ToInt32(BaudRateCmb. Text);
Serial Port1. DataBits = 8;
Serial Port1. StopBits=
System. IO. Ports. StopBits. One;
Serial Port1.Parity = System. IO. Ports. Parity. None;
Serial Port1. Open();
```

2) Sending Data

```
m_commands[0] = addressDevice;
m_commands[1] = command;
m_commands[2] = 0 ;
m_commands[3] = channel;
m_commands[4] = 0;
m_commands[5] = 1;
m_crc = CRC16(m_commands, 6);
m_commands[6] = m_crc[1];
m_commands[7] = m_crc[0];
Serial Port1. DiscardOutBuffer();
Serial Port1. Write(m_commands,0,
m_commands. Length);
```

3) Receiving Data

```
m_receive = new byte[Serial Port1.BytesToRead];
Serial Port1. Read(m_receive,0, Serial
Port1.BytesToRead);
if (m_receive.GetUpperBound(0) >= 6)
   {
      m_crc = CRC16(m_receive, 5);
      if (m_crc[0] == m_receive[6] && m_crc[1] ==
      m_receive[5])
       {
          returnValue = m_receive[3] * 256
                        +m_receive[4];
       }
      else
        {
          crcCount = crcCount + 1;
          lblCRC. Text = crcCount. ToString();
          return string. Empty;
        }
    }
  else
    {
       Serial Port1.DiscardInBuffer();
      return string. Empty;
    }
```

Implementation Method of the CRC Check

CRC value is calculated by the sending device and

saved in the end of the data frame. The equipment of receiving information recalculates CRC value. And take the CRC value to compare with the new CRC value that is calculated by the sending device based on the information that is returned by the receiving device. It means that there is something wrong with the communication if the two values are inconsistent.

(1) Set the register 0FFFFH which is CRC register.
(2) Save the result in the CRC register after the xor operation between the first byte of the data frame and low byte of the CRC register.
(3) Move the data to the right for a bit, and fill the high-order with '0', then detect the removing order which is lowest order.
(4) If the removing order is '0', return to step (3); if it is '1', xor operation between CRC register and a fixed value (0A001H) will be made.
(5) Repeat step (3) and (4) until 8 times. Then the 8 bits data can be gained.
(6) Repeat step (2) to (5) to deal with the next 8 bits data until all the bytes are finished to dispose.
(7) Now CRC register value is CRC value.

According to the above rules, the program of CRC check code in C# is as follows:

```
CRC16Lo = 0xFF;
CRC16Hi = 0xFF;
CL = 0x1;
CH = 0xA0;
 for (int dataInx = 0; dataInx < dataLength; dataInx++)
    {
      CRC16Lo = Convert.ToByte(CRC16Lo ^ data[dataInx]);
       for (int FlagInx = 0; FlagInx < 8; FlagInx++)
          {
            SaveHi = CRC16Hi;
            SaveLo = CRC16Lo;
            CRC16Hi =Convert.ToByte(CRC16Hi >> 1);
            CRC16Lo = Convert.ToByte(CRC16Lo >> 1);
             if ((SaveHi & 0x1) == 0x1)
               {
          CRC16Lo = Convert.ToByte(CRC16Lo | 0x80);
               }
             if ((SaveLo & 0x1) == 0x1)
               {
                 CRC16Hi = Convert.ToByte(CRC16Hi ^ CH);
                  CRC16Lo = Convert.ToByte(CRC16Lo ^ CL);
                 }
          }
        crc[0] = CRC16Hi;
        crc[1] = CRC16Lo;
    }
```

Note: The low in byte is in the front and the high byte of CRC16 is at the behind in Modbus-RTU Communication protocol.

Test Software Design

Use Visual Studio 2010 to achieve the design of software. The software interface of communication test program is shown in FIG.2.

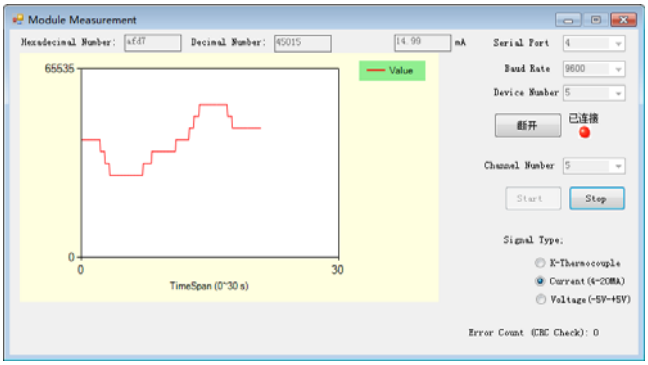

FIG. 2 THE SOFTWARE INTERFACE OF COMMUNICATION TEST PROGRAM

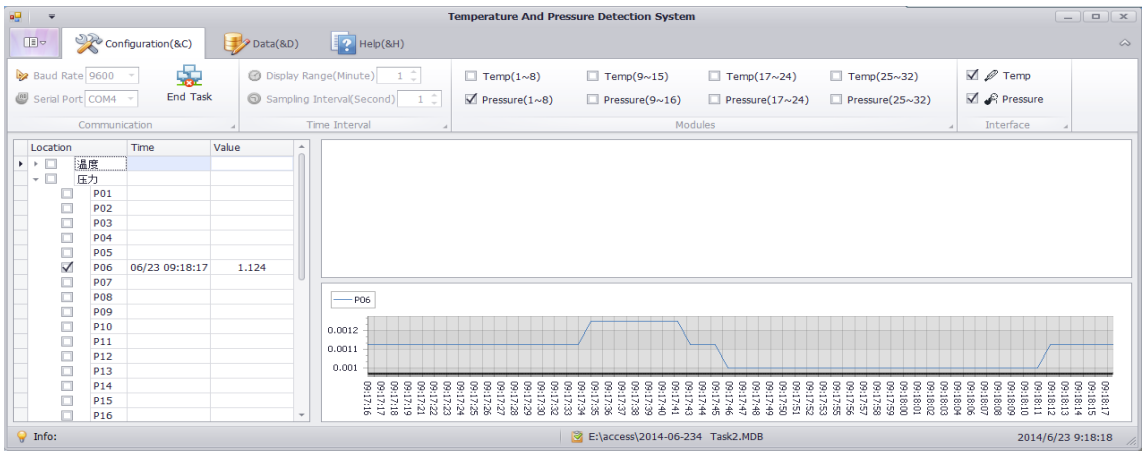

FIG. 3 THE APPLICATION SOFTWARE INTERFACE

Application Software Design

This project implements the communication between computer and ADVANCH 4018+ modules. The application software interface which is showed in FIG.3 makes the real-time display of the data measured by the modules with 64 channels in total both in node and chart possible. At the same time, the data that have been measured are saved to the database.

Conclusions

According to this project, serial port communication can be achieved based on Modbus-RTU communication protocol between computer and a few ADAM-4018+ modules that distribute data acquisition modules in C#.NET environment by Serial Port control. The result proves that the system is stable and can communicate in a timely manner. Serial Port control is easy and convenient to be used. For the realization of serial communication, Serial Port control will play more and more important role in different areas.

REFERENCES

Jiangquan Li, Hongtao Deng, Qiao Liu et al, Typical Examples of Serial Port Communication and Measurement and Control Application in Visual C#.NET, Publishing House of Electronics Industry, Beijing, 2012.

Lijun Dong, Shuwei Liu & Yiqing Hou,"Serial Communication Based on Modbus Protocol with Microsoft Visual Basic", Industrial Control Computer ,pp.8-11, August 2006.

Liping Li, Quanli Wei, Application of SerialPort Class in Messaging Software in .NET, Microcomputer & ITS Application, Vol.31, No.21, pp.11-13, 2012.

Wenquan Wang, Implementation of Serial Port Communication System Based on Serialport, Science Mosaic, pp.21-23, May 2011.

Yizhi Fan, Zhiyuan Chen, Dexuan Sun, Zhengfu Chen, Implementation of Serial and Parallel Communication Technology, Tsinghua University Press, Beijing, 2001.

Yizhi Fan, Zhiyuan Chen, Serial Communication Control between Visual Basic and RS-232, China Youth Publishing House, Beijing, 2002.

Yongji Guo, Xinggui Wang," Realization of the Communication between IPC and Intelligent Instrument Based on Modbus Protocol", Vol.20, No.1, pp.102-104, 2008.

Yuefei Wu, Implementation of Serial Port Communication between Upper Machine and PLC by SerialPort, Internet Fortune, pp.156-157, January 2009.